



Intel® Extended Server Memory Architecture (ESMA): Overcoming the 4 GB Memory Barrier for IA-32

Information for Software Developers

Version 6.0 11/1/99

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

The Intel® Pentium®, Pentium® Pro, Pentium® II, Pentium® II Xeon™, Pentium® III, and Pentium® III Xeon™, processors may contain design defects or errors known as errata. Current characterized errata are available upon request.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference www.intel.com/procs/servers/performance or call (U.S.) 1-800-628-8686 or 1-916-356-3104.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725. (If you are located outside the United States please call 303-675-2120.) Information may also be obtained by visiting Intel's web site at <http://www.intel.com>.

Copyright © 1999 Intel Corporation. All rights reserved. *Third-party brands and names are the property of their respective owners.



283872-001

Table of Contents

Executive Summary	1
ESMA Overview	2
Introduction and Benefits.....	2
ESMA Technology Elements	3
Intel® Core Processor 36-bit Modes	3
Cache Support.....	4
Chipset Support.....	4
Summary.....	5
ESMA Usage Examples	5
Database Cache: Oracle8* Usage Example1	5
The SAS* System: SAS Institute Usage Example	6
SAP <i>live</i> Cache Usage Example	7
Summary.....	8
Operating Systems Support for ESMA	9
Broad-based Support.....	9
Windows NT* Server Support for ESMA	9
Intel® PSE36 Driver and SDK	10
Application Modification Requirements for ESMA.....	10
Windows 2000 ESMA Support.....	11
UnixWare* 7 Support for ESMA	12
Solaris* 7 Operating Environment Support for ESMA	13
Linux* Support for ESMA	13
DYNIX*/ptx Support for ESMA	13
Summary.....	14
References.....	14

Executive Summary

The Intel® Extended Server Memory Architecture (ESMA) is a collection of technologies that transcend the 4 gigabyte (32-bit) memory barrier for enterprise applications. Server platforms using Intel® Pentium® II and Pentium® III Xeon™ processors provide complete support for this architecture. ESMA support is also planned for future Intel® 32-bit processors.

The Intel ESMA includes technologies that provide full 36-bit addressing support from the processor, Level 1 and 2 cache, and the chipset. Together, they offer a non-intrusive, evolutionary path for enterprise applications to exploit more than 4 GB of memory, producing unsurpassed performance, cost/performance and scalability for applications running on Intel® Architecture-based servers.

Applications that utilize the Intel ESMA can enable information technology (IT) managers to realize greater competitiveness for their business-critical applications and solutions. Applications that can benefit from ESMA include:

- Database management systems
- On-line transaction processing
- Data warehousing
- Decision support systems
- Internet infrastructure servers (such as Directory and Proxy/Cache)
- Line-of-business applications
- Other end-to-end business solutions

In one benchmark study¹, ESMA improved the elapsed time for critical Oracle8* queries by two to four times.

The majority of leading operating systems and enterprise applications available today already support the Intel Extended Server Memory Architecture or have announced plans to support it.

This document is designed to give software engineering and product marketing managers a brief overview of Intel ESMA technology and its benefits in order to help them understand how they might take advantage of ESMA in their software products. This paper also provides an overview of operating system and tools support for ESMA.

ESMA Overview

Introduction and Benefits

With increasing frequency, databases, decision support systems and other end-to-end business solutions are growing larger than the 4 GB memory limitation imposed by 32-bit addressing.

To meet the needs of enterprise applications and to provide headroom for next-generation business solutions, Intel developed the Extended Server Memory Architecture (ESMA), enabling servers based on the Intel Pentium II and Pentium III Xeon processors to provide 36-bit addressing and thus to support up to 64 GB of main memory. The Pentium III Xeon processors also incorporate new memory streaming instructions that speed memory accesses and amplify the performance advantages of ESMA.

Intel ESMA enables enterprise applications to hold all or a larger portion of a customer's enterprise databases or data sets in main memory (DRAM) rather than having to move that data repeatedly to and from the disk subsystem. Since disk transfers are several orders of magnitude slower than data transfers between the processor and main memory (MB/sec vs. GB/sec, respectively), ESMA can significantly increase an application's performance and throughput and reduce application access latencies. In addition, fewer disk I/O operations also lead to lower CPU utilization by the disk driver, thus further enhancing application performance. These benefits apply to many medium-size enterprise databases or data sets in the gigabyte range as well as to very large databases (VLDB) and data sets in the hundreds of gigabytes to terabytes range.

For example, on-line transaction processing (OLTP) applications can complete business transactions more quickly. Early testing indicates gains of 10-20% on industry-standard TP (transaction processing) benchmarks such as TPC-C when memory size is increased from 4 to 8 GB (with everything else being equal). An application's exact acceleration will vary depending on the size of the customer's database, system configuration and the characteristics of the application.

In enterprise decision support systems (DSS), the added performance can result in faster queries. Other examples of business-critical applications where ESMA can deliver greater throughput and faster response time include line-of-business (LOB) solutions, data warehousing applications and Internet Proxy/Cache and Directory Services.

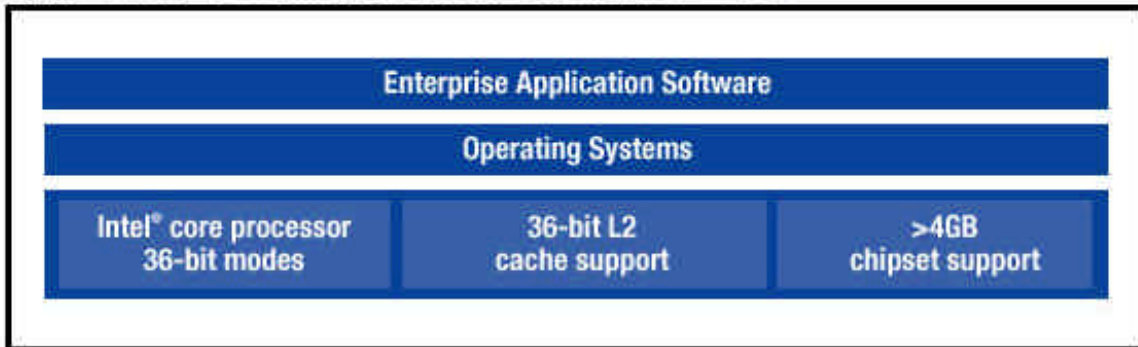
The following vendors and products are just a few of those that support the Intel® Extended Server Memory Architecture:

- Microsoft's SQL Server*
- Oracle's Oracle8* database technology
- SAP's *liveCache**
- SAS* Institute's SAS System's suite of data warehousing and decision support software

ESMA Technology Elements

Intel ESMA overcomes the 4 GB limitation by combining technologies in the processor, the Level 1 (L1) and Level 2 (L2) caches and server chipsets, such as the Intel® 82450NX chipset (Figure 1).

Figure 1: Key Technology Elements of the Intel® ESMA



Intel® Core Processor 36-bit Modes

The Intel ESMA has two 36-bit addressing modes in the core processor that offer different processor addressing schemes depending on the operating system in use:

- PAE – PAE stands for Physical Address Extensions and was introduced with the Intel® Pentium® Pro processor in 1995.
- PSE36 – PSE36 is an acronym for 36-bit Page Size Extension and is a new feature of the Pentium® II and Pentium® III Xeon™ processors. The PSE driver enables applications running under the Microsoft® Windows NT® 4.0 operating system to overcome Windows NT's 4 GB memory limitation. The Pentium II and Pentium III Xeon processors maintain compatibility with PAE.

Table 1: Intel® Processor Support for ESMA

Processor	ESMA support
Pentium®	No
Pentium® II	No
Pentium® Pro	PAE36 only
Pentium® II Xeon™	Yes
Pentium® III Xeon™	Yes
Future Intel® 32-bit processors	Yes

(Note, Pentium II processor is capable of ESMA, but the external chipset controlling the cache must also support ESMA; typically, chipsets associated with the Pentium II are incapable of dealing with ESMA.)

Full technical descriptions of these 36-bit addressing modes can be found at <http://developer.intel.com/design/PentiumII/manuals/243690.htm>. For specific operating system support, see Table 3 in the Operating Systems Support for ESMA section of this document.

Both modes allow operating systems to support up to 36-bit addressing or 64 GB of main memory (DRAM) and therefore provide significant headroom for enterprise applications.

Cache Support

ESMA includes high-performance L2 processor caches with full support for 36-bit (64 GB) memory addressability. Since caches are designed to hold frequently used data close to the processor, they must also store the 36-bit address of the data being cached. Note that the memory addressability of the cache is not to be confused with the size of the cache (how much data the cache can hold). As Table 2 shows, Pentium Pro, Pentium II and Pentium III Xeon processors include L2 caches with 36-bit memory addressability, and hence are processors that support the Intel Extended Server Memory Architecture.

Chipset Support

ESMA includes chipsets that allow OEMs to design systems that can be fully configured with more than 4 GB of DRAM. Along these lines, the Intel 82450NX chipset, which complements the Pentium® II and Pentium® III Xeon™ processors, supports up to 8 GB of DRAM today, and the amount of DRAM supported by future chipsets is expected to grow. High-end OEM versions of chip-sets, for example those in the Sequent NUMA-Q 2000 systems, already support 64 GB of RAM now. Applications that support ESMA will obtain transparent benefits from these increases in memory over time.

Table 2: L2 Cache Support for ESMA

	L2 Cache Addressability	L2 Cache Size	ESMA Support
Pentium® II	4 GB (32-bit)	512 KB	No
Pentium® Pro	64 GB (36-bit)	512 KB, 1 MB	Yes
Pentium® II Xeon™	64 GB (36-bit)	512 KB, 1 MB, 2 MB	Yes
Pentium® III Xeon™	64 GB (36-bit)	512 KB, 1 MB, 2 MB	Yes

Summary

The Intel® ESMA provides technologies in the processor, chipset and caches that transcend the 4 GB (32-bit) memory barrier and enable applications to address up to 64 GB of memory. Server platforms that use the Intel Pentium II and Pentium III Xeon processors provide hardware support for this architecture.

ESMA enables enterprise and Internet applications to increase performance and throughput by using the memory beyond 4 GB to store large amounts of data. Accessing this data in memory is much faster than disk access and results in lower application access latencies. It can also significantly reduce I/O traffic, further accelerating application performance.

ESMA Usage Examples

Oracle* Corporation's Oracle8* application has achieved a two- to fourfold improvement in key operations by taking advantage of the Intel ESMA. SAS* Institute's SAS System* for Windows NT* will utilize ESMA beginning with Version 7. SAP also implemented ESMA in order to increase the performance and throughput of *liveCache*.

Database Cache: Oracle8* Usage Example

Oracle has modified its Oracle8 Server code-base to exploit the Intel ESMA and realized dramatic improvements in the speed with which database information is accessed and processed.

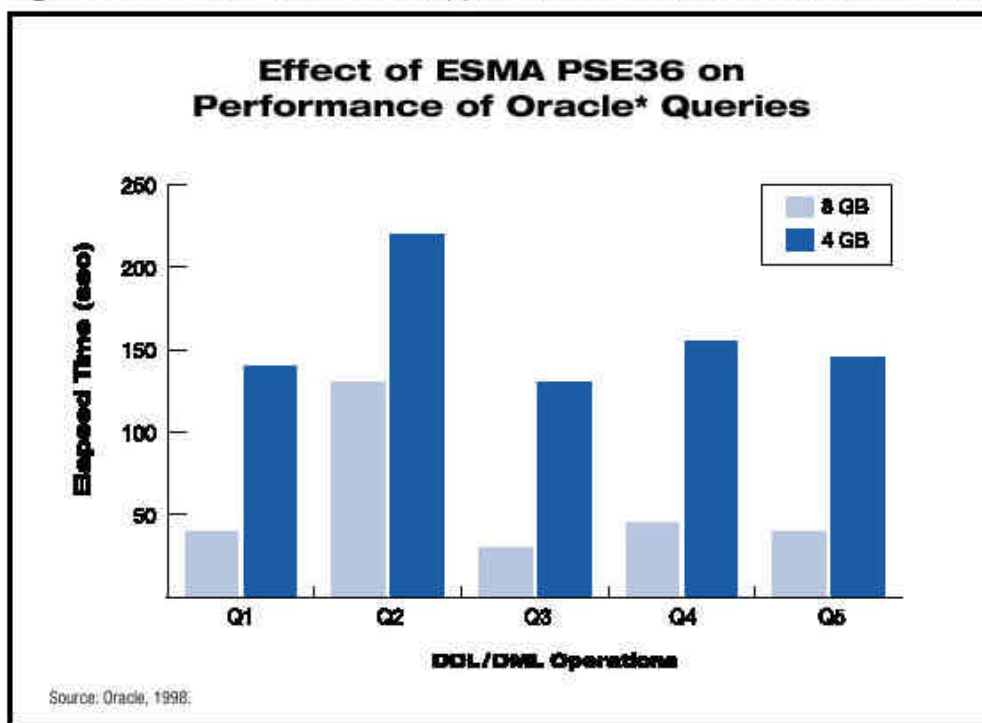
ESMA enables Oracle8 to overcome the Windows NT Operating System's 4 GB memory limit and thereby support larger databases and more users with increased performance. Without ESMA support, a maximum of only 3 GB was available for use by Oracle8 code, database user connections and Shared Global Area (SGA). With ESMA, an additional 4 GB of physical memory is available to Oracle8 SGA. Oracle8 uses it as a RAM disk to cache blocks of data from the disk, thus greatly reducing slow disk I/O transfers and significantly improving database performance.

Figure 2 shows the results of early tests. In these tests, an additional 4 GB of physical memory data cache available to Oracle8 Shared Global Area (SGA) improved Oracle8 Server query response time by two to four times. Early benchmarks also indicate gains of 10-20% on industry-standard TP benchmarks (e.g., TPC-C) when memory size is increased from 4 to 8 GB (with everything else being equal).

Oracle8 was tested on a server with the 400 MHz Pentium II and Pentium III Xeon processors and 1 MB of L2 cache, running Windows NT Server 4.0, Enterprise Edition with the PSE36 driver. The Oracle8 database used for the test contains two tables, the first with 3 million rows and the second with 1,000 rows. The database was larger than 4 GB, but smaller than 8 GB. The database with the Intel ESMA support was fully cached. The elapsed time on the server was measured for the following DDL/DML operations in both test cases:

- **Q1:** Full table scan on the 3 million row table
- **Q2:** Index creation on the 3 million row table
- **Q3:** Numerical computation (SUM function) on one field of the 3 million row table
- **Q4:** Two-way join between the 3 million row table and the 1,000 row table
- **Q5:** Insert into a new table using data from the 3 million row table

Figure 2: Effect of ESMA PSE36 Support on Performance of Oracle8 Queries*



For a detailed Oracle* white paper, Oracle8* Support for the Intel® Extended Server Memory Architecture: Achieving Breakthrough Performance, visit: <http://www.intel.com/procs/servers/isv/oracle/esma.htm>.

The SAS* System: SAS Institute Usage Example

Beginning with Version 7 of The SAS* System for Windows NT* 4.0, ESMA capabilities are available to users of the SAS System's suite of data warehousing and decision support software. Through SAS* Institute's support for this architecture, SAS applications can make direct use of memory beyond the 4 GB limit, to greatly improve access for various types of data processing operations.

Initial tests show significant performance improvements for some types of operations that make use of ESMA, particularly those in which access to large amounts of disk-based data are required. In these cases, when much or all of the data can reside in very fast memory (relative to extremely slow disk drives), the central processing power can be more effectively utilized.

In addition to lowering or eliminating the overhead of waiting for disk systems to handle large data requests, ESMA has several other benefits when it is used effectively:

- When information is retrieved through ESMA, it can dramatically decrease disk channel traffic, resulting in greater performance of other operations that may be occurring on a server system.
- Since ESMA can, in some cases, be immune to disk bottlenecks, main SAS System processing throughput can remain largely unaffected by other server activities, as long as central processing power is available.
- SAS System ESMA operations bypass the Windows NT Server 4.0 disk cache, leaving it available to handle other application requirements on a server system.
- The amount of memory available to SAS System processing is limited only by hardware availability, up to the ESMA limit of 64 GB of main memory. This can be far greater than typical applications can access through standard access in Windows NT Server 4.0, which is limited to 3 GB per application. When these large amounts of memory are available, very impressive performance improvements should result.

Look for a forthcoming white paper covering SAS application specifics and ESMA. For information, check: <http://www.sas.com/partners/enterprise/msinfo/intro.html>.

SAP* *liveCache: In-Memory Database Usage Example**

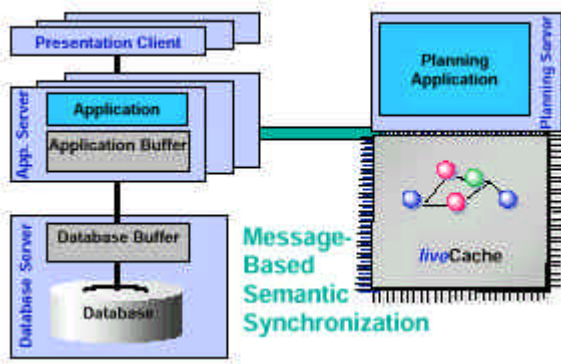
In today's competitive environment, the ability to respond to customer requests with speed and precision is vital to success. These intense customer demands have placed extraordinary pressure on organizations' planning and operational systems.

SAP's *liveCache* is an extremely fast memory-based technology for executing sophisticated, data-intensive functions. Operating within the SAP Business Framework, *liveCache* enables entirely new class of applications by offering orders of magnitude performance improvements for functions such as forecasting and supply chain planning and optimization. For example, *liveCache* makes it possible for a sales representative to perform comprehensive available-to-promise calculations during a phone conversation with a customer.

SAP has implemented *liveCache* using ESMA on Windows NT Server 4.0. Using the ability to access data in memory with ESMA, SAP's *liveCache*, a memory-based cache technology, fulfills the following requirements imposed by advanced planning and optimization techniques (see Figure 3).

- Provide up-to-date data
- Minor performance impact on transactional processing
- Concurrency and transactional behavior supported
- On the fly reconstruction of data structures

Figure 3 SAP's *liveCache* – Memory-based cache architecture



In supporting today's business applications, superior performance is one of the key requirements. Accessing large databases via traditional replication techniques as illustrated in Figure 3 using slow, disk-based database processing presents problems in both performance and accuracy. ESMA allows SAP to remove the 4GB-memory barrier, and implement the memory-based cache architecture as illustrated in Figure 3. (As an example, holding 500,000 orders simultaneously in main memory, where each order has an average of 100 components, 6 productions steps, 50,000 materials to select from, and 5,000 resources to dispatch, can require up to 10GB of main memory.)

By administrating all data in very large main memory, access to data is greatly accelerated with respect to hard-disk I/O. Access to data via a remote database server has a minimum footprint of 1000 microseconds, whereas today's applications require an average access time for a single data item of less than 10 microseconds.

More information on SAP *liveCache* can be found at <http://www.sap.com>

Summary

Applications that take advantage of the Intel ESMA extended memory support can gain significant performance advantages with relatively simple modification to the applications. Given the 64 GB headroom of the Intel ESMA, the amount of DRAM supported by future Intel® 32-bit processor-based Standard High Volume (SHV) servers can be expected to grow beyond 8 GB over time. Applications such as database management systems, on-line transaction processing, data warehousing, decision support systems, Internet infrastructure servers (such as Directory and Proxy/Cache) and other data-intensive business applications will obtain even greater performance benefits and transparent growth from these increases in memory over time.

Table 3: Operating System Support for ESMA

Operating System	ESMA Support	Status
Windows NT* Server Enterprise Edition 4.0, Service Pack 3.0 or later	Yes	PSE36 driver must be included in the OEM server platform, or available from Intel at http://developer.intel.com/design/perftool/pse36
Windows* 2000 Server, Enterprise Edition	Yes	Available upon release
UnixWare* 7 (release 7.0 & 7.1)	Yes	Currently Available
Solaris* 7 Operating Environment 3/99	Yes	Fully supported, no additional driver required
Linux 2.2	Forthcoming	Targeted for release in 12/99
NetWare*	Forthcoming	Targeted for release (“6-Pack”) in 2000
DYNIX*/ptx 4.4.4	Yes	Currently available. 4.4.4 supports 64GB of RAM on a 64 processor system. Future release 4.5.0 will support 64GB of RAM on a 32 processor system.

Operating Systems Support for ESMA

Broad-based Support

Support for Intel Extended Server Memory Architecture is available for Microsoft's Windows NT* Server 4.0 and Windows* 2000 Server, SCO's UnixWare* 7, Sun's Solaris* 7 Operating Environment 3/99, and an upcoming release of Linux* and Novell's NetWare*. Other OEM versions of Unix, such as Sequent's DYNIX*/ptx 4.4.4, already support Intel ESMA and 64 GB of RAM on 64-processor systems.

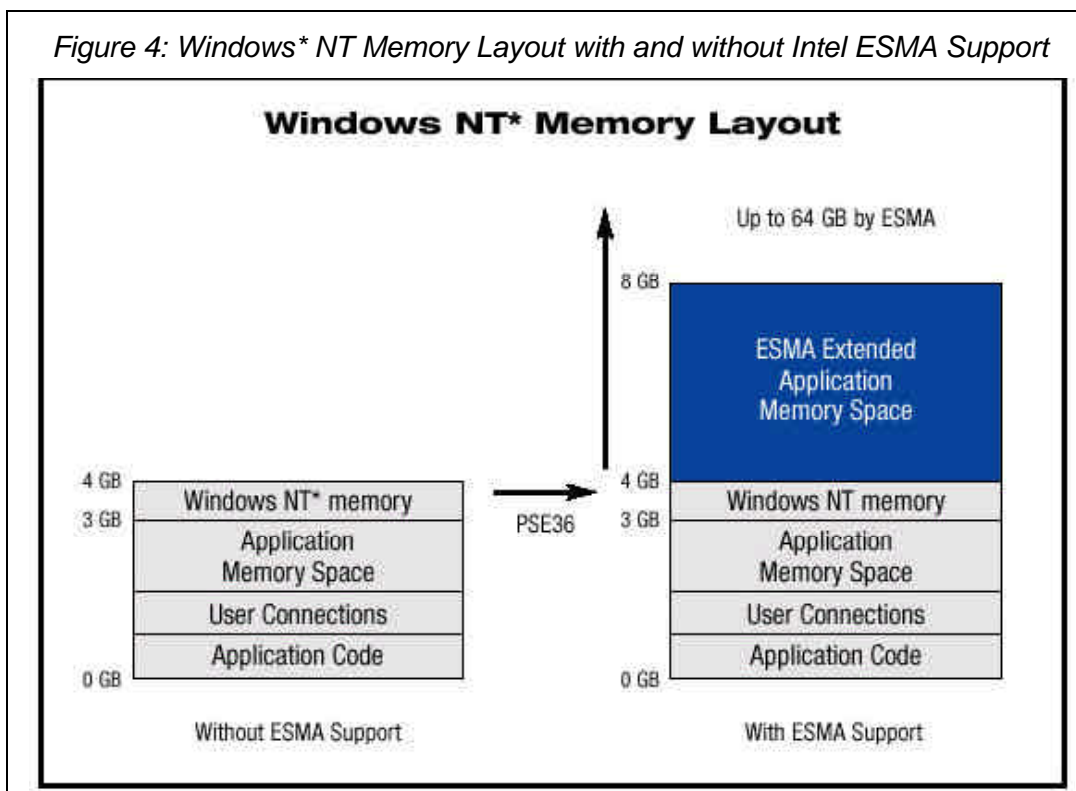
Current hardware requirements for operating systems that support Intel® ESMA are: Intel® Pentium® II or Pentium® III Xeon™ processor-based servers, a 36-bit memory controller such as that provided by the Intel® 82450NX chipset, and more than 4 GB of system RAM.

Windows NT* 4.0 Server Support for ESMA

Microsoft Windows NT Server 4.0* does not recognize memory beyond 4 GB. The Windows NT Server Operating System 4.0 supports Intel ESMA via the use of the PSE36 driver, which interfaces directly to the processor's PSE36 36-bit addressing mode and manages the use of memory above 4 GB.

The PSE36 driver is a standard RAM disk device (based on the Windows NT DDK RAM disk driver) that lacks a file system and is backed by memory that is unused by the operating system. This memory optionally exists at physical addresses above 4 GB. The PSE36 driver device functions very much like a raw disk partition, except that the latency is much lower than the latency for a raw disk partition.

Figure 4: Windows* NT Memory Layout with and without Intel ESMA Support



Only one process may open the PSE36 driver and use it at any given time. This process gets exclusive use of the device and the entire RAM disk when it opens the device. Applications cannot directly map memory beyond 4 GB. Only the PSE36 device driver is aware of the memory it controls. RAM disk space cannot be shared between processes; it is never mapped into the address space of any process, and it is not backed by the Windows NT Server 4.0 page file.

The PSE36 driver is designed to be multithread safe. Multiple threads on multiple processors can simultaneously copy data to and from the extended memory space. Address space (4 MB) in the kernel's address map is mapped dynamically onto the memory controlled by the PSE36 device on a per-processor basis. The requested buffer (typically 2-4 KB) is copied, and the mapping is then deleted. This improves performance because a system-wide Translation Look-ahead Buffer (TLB) shoot-down (invalidation) would require interrupting all processors. By dynamically mapping on a per-processor basis, each processor needs to invalidate only its own TLB.

Intel® PSE36 Driver and SDK

For Microsoft's Windows NT Server Enterprise Edition 4.0, Intel supplies and supports this driver via server OEMs. (Server OEMs integrate the driver into their Pentium® II and Pentium® III Xeon™ processor-based server platforms and deliver the complete solution to end-users and system integrators.)

For early developers, the PSE36 driver and Software Developer's Kit (SDK) are also available free of charge from Intel. The Intel® PSE36 SDK runs on Pentium II and Pentium III Xeon processor-based systems running Microsoft* Windows NT* Server Edition 4.0 (with Server Pack 3 or later.) The PSE36 SDK contains the PSE36 driver binary, API specification and include file, sample programs with source code, tests, performance monitoring facilities, and documentation. Applications can be built using any Win32 compatible C-compiler in this environment. For further information, refer to <http://developer.intel.com/design/perftool/pse36/>

Application Modification Requirements for ESMA

For applications in Windows NT Server 4.0 to take advantage of the PSE36 address space, they must be modified. In most cases, these modifications place some burden on the application for managing the memory in the PSE36 address space to copy from or to memory controlled by the PSE36 device. Simple-to-use APIs are available and can be used by many applications with a relatively small amount of effort, since applications need only make changes related to memory management, such as shared buffer pool management. Data must be copied back and forth between the PSE36 address space and the application's virtual address space. Applications that are already designed to use disk space as an extended data storage or cache, and therefore to copy data between the disk and the user's virtual address space, should need minimal modifications.

The application is also responsible for data integrity. Since the PSE36 driver is designed to be multithread safe, multiple threads can copy data to and from the extended memory space simultaneously. No serialization is required outside of the driver to limit access. However, the PSE36 driver does not prevent two threads from reading and writing to the same buffer at the same time. As a result, the applications must deal with these potential conditions to maintain data integrity.

I/O to and from more than 4 GB memory is not supported. Application I/O to and from memory beyond 4 GB is accomplished by first having the PSE36 driver copy data to memory below 4 GB and then completing I/O. The pages beyond 4 GB must be 4 MB in size, although pages in the range of 0-4 GB can be either 4 KB or 4 MB in size. The memory controlled by the PSE36 device is non-pageable. This means that an application is responsible for managing memory controlled by the PSE36 driver.

These limitations arise from the fact that Windows NT Server 4.0 has no knowledge of the memory above 4 GB. But for applications that are already managing large data sets, data cache and shared buffer pools, PSE36 APIs provide a simple way to extend application access to physical memory beyond 4 GB on Pentium II and Pentium III Xeon processor-based servers.

Windows 2000 Support for ESMA

In contrast to Windows NT Server 4.0, Windows 2000 provides native PAE support. With Windows 2000, PAE support is optional; two sets of kernels are provided – one set is used when 4GB or less of physical memory is present, and the other set is used when more than 4GB of physical memory is present. The use of either kernel can be forced via boot switches. Whereas non-PAE kernels only allow for applications to exist within the first 4 GB of physical memory, the PAE kernels allow for applications to exist throughout physical memory upto 64GB in size. In both sets of kernels, applications are still provided with a 4GB virtual address space. Typically, a Windows 2000 application's 4GB virtual address space is broken into 2 parts: 2GB address space for the user, and 2GB address space for the kernel. (Note: Windows NT Server 4.0 Enterprise and various advanced versions of Windows 2000 provide a switch allowing for 3 GB user address space, and 1GB kernel address space.) Applications executing under the PAE kernel that need more physical memory than 2GB (or the optional 3GB) are provided with Microsoft's Address Windowing Extensions (AWE) APIs in order to utilize physical non-paged memory. AWE APIs allow for applications to map and unmap individual pages of additional physical memory into a special region of the application's virtual address space. Diagrams regarding the memory layout of applications and Windows 2000 are detailed in Microsoft's white paper concerning ESMA support, and are available at: <http://www.microsoft.com/HWDev/NTDRIVERS/AWE.htm>.

Application Modification Requirements for ESMA

Developers should note that utilizing the AWE APIs is significantly different from utilizing the PSE36 driver. In order to utilize ESMA, applications previously developed that use the PSE36 driver must be modified. With the PSE36 driver, the kernel is unaware that the memory above 4GB exists – only the PSE36 driver has any knowledge of this memory. Therefore, applications cannot map or directly access this additional memory. With PSE36, data needing to be placed in the additional memory must be taken from memory in the application's address space and "written" to the PSE36 device. However, with the AWE APIs, since physical memory is actually mapped into an application's virtual address space, any changes to the mapped region take effect immediately. The fundamental difference between the AWE APIs and the PSE36 driver is that the former is utilizing *memory that has been mapped into an application's virtual address space*, whereas the latter functions like a *disk-device*. Developers seeking more information can also consult Microsoft's document at the URL below for further information:

<http://www.microsoft.com/HWDev/NTDRIVERS/AWE.htm> or the latest Windows 2000 SDK.

UnixWare* 7 Support for ESMA

SCO's UnixWare* 7 Operating System uses Intel® ESMA to support up to 64 GB of physical memory. Large memory support is implemented through the use of ESMA 36-bit addressing mode PAE page-table entries. UnixWare 7 provides a new interface, called Dynamically Mapped Shared Memory (DSHM), which is an extension to the traditional shared memory interface. By taking advantage of ESMA, DSHM allows a single application to use large amounts of physical memory.

In UnixWare 7, memory is configured either as dedicated or general-purpose, depending on its intended use. Dedicated memory is used for large shared memory segments. System administrators can configure any amount of dedicated memory, up to the limits of physical memory. Dedicated memory can be used for large shared memory segments created and accessed through either the traditional Shared Memory (SHM) interfaces or the newer DSHM interfaces.

With the introduction of UnixWare 7 release 7.1, as much as 16 GB of general-purpose memory can be configured. This increase in general-purpose memory can benefit a variety of workloads. For example, system paging or swapping may be reduced or eliminated for workloads whose total working set is larger than 4 GB. As a second example, since UnixWare* 7 uses general-purpose memory to cache file pages, the use of ESMA for general-purpose memory may benefit file servers by providing a larger cache, thereby reducing disk activity and increasing both throughput and response time for such machines.

While UnixWare 7 ESMA support allows systems to use up to 64 GB of physical memory, traditional interfaces effectively limit the amount of physical memory that can be accessed by a single user process because of virtual addressing limitations.

Using ESMA, DSHM provides the ability to create shared memory segments up to the limits of physical memory. A process can then establish and dynamically change mappings between its 3 GB virtual address space and different pages of a DSHM segment.

ESMA and DSHM can allow applications to realize significant performance gains. For example, database programs typically create shared memory segments for use as buffers to cache data in memory for rapid access. DSHM allows more data to be cached in memory, thereby reducing data access time for a larger data set.

For details on the DSHM interfaces, including constraints placed on applications which use them, see <http://uw7doc.sco.com/> and search for “DSHM.”

Solaris* Support for ESMA

Sun's Solaris* 7 Operating Environment now supports the Intel® ESMA through dynamically loaded memory management subsystems and a PAE memory management subsystem.

Solaris offers the ability to dynamically load and bind the appropriate memory management subsystem at boot time; this also allows the system administrator to override the memory management subsystem chosen by the system. The support for dynamically loaded memory management subsystems is provided, although only one such subsystem exists today, the non-PAE 32-bit subsystem.

The Solaris 7 Operating Environment supports a PAE memory management subsystem. This allows all memory up to 32 GB to be used as general-purpose memory for application memory, kernel buffers, file system and buffer caches, streams buffers, and so forth. No application changes are needed to make use of the additional memory, nor is there a need to install any special version of the Solaris Operating Environment. Depending on the amount of memory present in the system at boot time, the appropriate memory management subsystem will be loaded and used with no user intervention.

The Solaris 7 PAE memory management system does not require additional device drivers, since the Solaris Device Driver Interface (DDI) inherently allows the specification of I/O above 4 GB. However, if device controllers limited to DMA below 4 GB are used, an additional memory copy may be required that would reduce the benefit from the additional memory. Solaris uses heuristics to allocate pages with limited I/O above 4 GB, but this does not necessarily eliminate all such I/O. It is recommended that systems have PCI device controllers that support the Dual Address Cycle (DAC) in both hardware and the Solaris device driver to eliminate this overhead. Contact the system vendor and/or device vendor for details on their support for DAC.

For details on Solaris 7 Operating Environment 3/99, see <http://www.sun.com/solaris/>

Linux* Support for ESMA

Intel is working with the Linux* community to ensure that ESMA will be supported in a future Linux release of the 2.2 kernel scheduled for December of 1999.

DYNIX*/ptx Support for ESMA

Sequent*'s DYNIX*/ptx Operating System supports Intel ESMA via direct use of PAE page-table entries, and has done so since Sequent NUMA-Q 2000 systems, based on the Pentium® II processor, first shipped in 1996. Those systems shipped with 16 processors and 16 GB of RAM and ran DYNIX/ptx Version 4.4.0. Sequent DYNIX/ptx Version 4.4.4 now supports 64 GB of RAM on a 64-processor system. In 1999, Sequent expects to support 64 GB of RAM on a 32-processor system running DYNIX/ptx Version 4.5.0.

With the support of PAE page-table entries, DYNIX/ptx has full knowledge of the memory beyond 4 GB. Applications need not be aware that they are running in memory beyond 4 GB, so no special device drivers, APIs or application modifications are needed. A binary compiled on a pre-PAE version of DYNIX*/ptx runs unmodified on a PAE version of DYNIX/ptx, and can take full advantage of physical memory above 4 GB.

The virtual address layout of user processes is unaffected by PAE support. There are slight differences in the kernel address space due to the fact that PAE page-table entries are twice as large as traditional page-table entries.

Any process may be assigned physical memory from within the full 64 GB physical address space. A process may make full use of any page of memory assigned to it, regardless of whether the page resides above or below the 4 GB boundary.

All disk devices may perform DMA operations directly from any portion of the 64 GB of physical memory; no copying or DMA mapping is required. Some network controllers can perform DMA operations only from the lower 4 GB of physical memory. The driver handles any copying or mapping required to perform the network transmission and reception; user processes need not concern themselves with this limitation.

Summary

The Intel® Extended Server Memory Architecture combines technologies in the processor, chipset and caches to enable servers based on the Intel® Pentium® II and Pentium® III Xeon™ processors to support 36-bit memory addressing and thus to utilize up to 64 GB of memory. By designing their applications to take advantage of this capability, software developers can enhance the performance, throughput and accuracy of their applications.

Intel ESMA is widely supported by leading operating systems. A PSE36 driver, available from Intel or provided by server OEMs, enables Windows* NT Server 4.0 applications to utilize ESMA. Windows2000 provides support for ESMA through its AWE API. Various versions of the Unix* operating system can utilize ESMA with few or no modifications.

References

1. Oracle8* Support for the Intel® Extended Server Memory Architecture: Achieving Breakthrough Performance, Oracle whitepaper, June 1998. (<http://www.intel.com/procs/servers/isv/oracle/esma.htm>)
2. Intel® Memory Streaming Instructions: Increasing the Performance of Server Applications, Intel whitepaper, 1999. (<http://developer.intel.com/design/PentiumIII/papers>)
3. Intel® Extended Server Memory Architecture, Intel whitepaper, 1998. (<http://developer.intel.com/>)
4. Intel® Architecture Software Developer's Manual, Volume 3: System Programming Guide Addendum, Intel Order Number: 243690-001
5. Intel® PSE36 Driver and SDK Reference Manual, 1998. (<http://developer.intel.com/design/perftool/pse36/>)
6. The SAS Institute ESMA White Paper, 1999. (<http://www.sas.com/partners/enterprise/msinfo/intro.html>)